On-The-Fly TLB Generation to Realize Variable Page Size Support on Linux / IA64

Gelato ICE 2007
Itanium Conference & Expo
Kernel Track

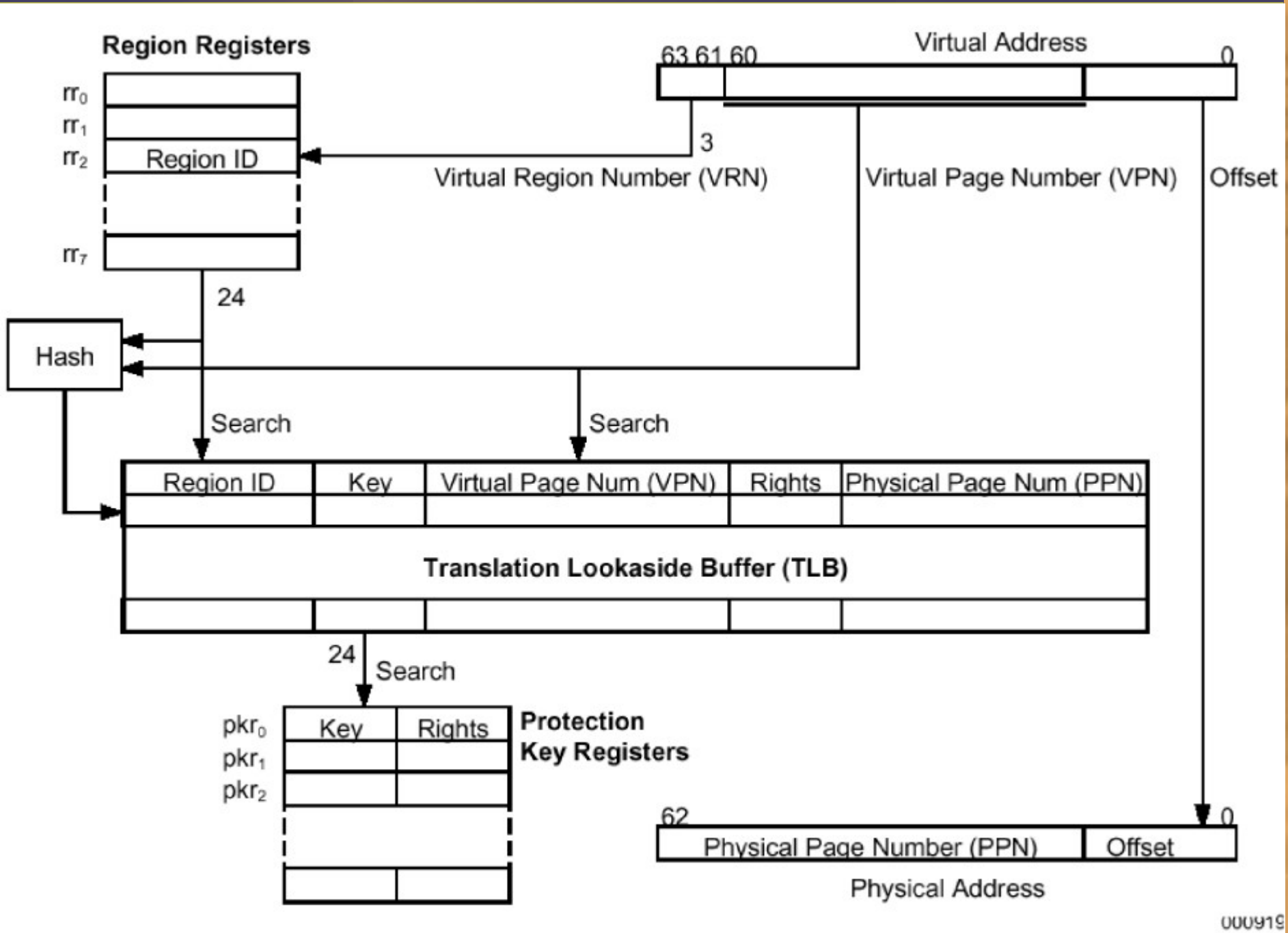
2007-04-17

Christoph Lameter
clameter@sgi.com
SGI
High Performance Computing

Overview

- TLB handling on Itanium Processors
  - Paging and TLBs
  - Memory Regions
  - VHPT lookups short and long
- Linux/IA64
  - Roles of Regions
  - VMALLOC area
  - Memory map
- IVT and the exception handlers
- A programmable TLB generator
- Performance numbers
- Conclusion

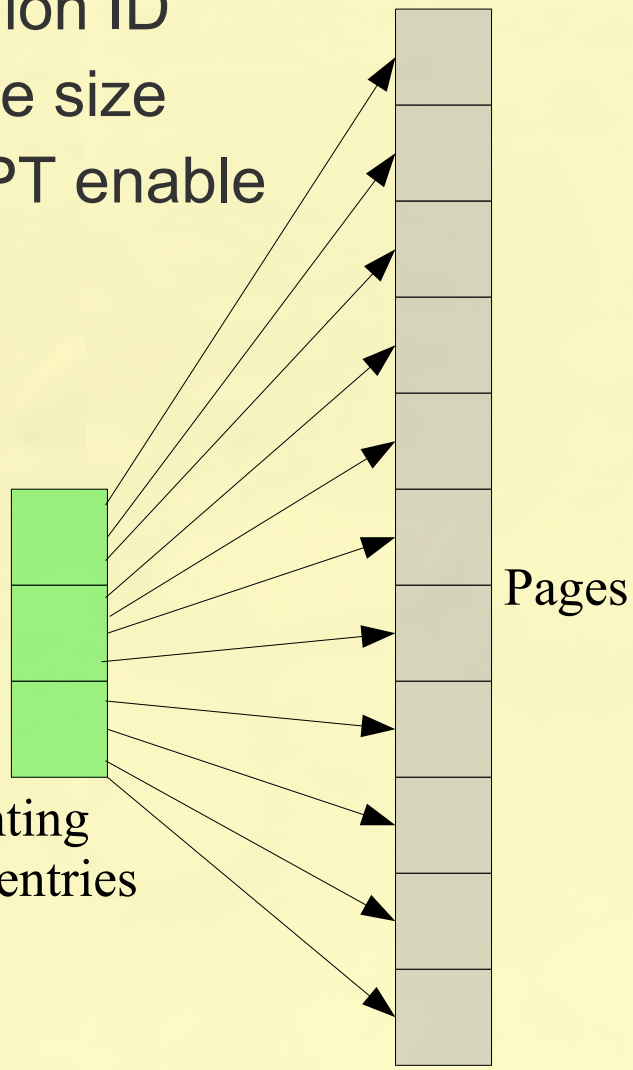
Paging and Translation Lookaside Buffers



Itanium Regions & Page Tables

- 7 Regions
- Two modes
- Short VHPT: One page size per region
- Long VHPT: page size encoded in page table
- Short VHPT -> VMLPT address
- Page table “walking”

- Region Register
  - Region ID
  - Page size
  - VHPT enable



VHPT lookup address pointing to an array of short VHPT entries commonly known as PTEs

TLB handling on Itanium

- TLBs generated via exception handlers or VHPT lookups
- 128 L2 TLB entries that are able to support all page sizes
- 32 L1 TLB entries supporting 4K page size only
- Page tables and translation entry lookups (VMLPT)

Programmable Exception Handler

- TLB insertion can be completely controlled via exception handlers
- IA64 only modifies PTEs through exception handlers of the OS. MMU does not set bits.
- No support for page tables in the classic sense unless realized by exception handlers.
- Exception handlers set dirty bit, accessed bit etc.
- Separate handlers for instruction and data TLBs
- No vector table instead a certain number of bundles (64, 16) is reserved at a particular memory location.

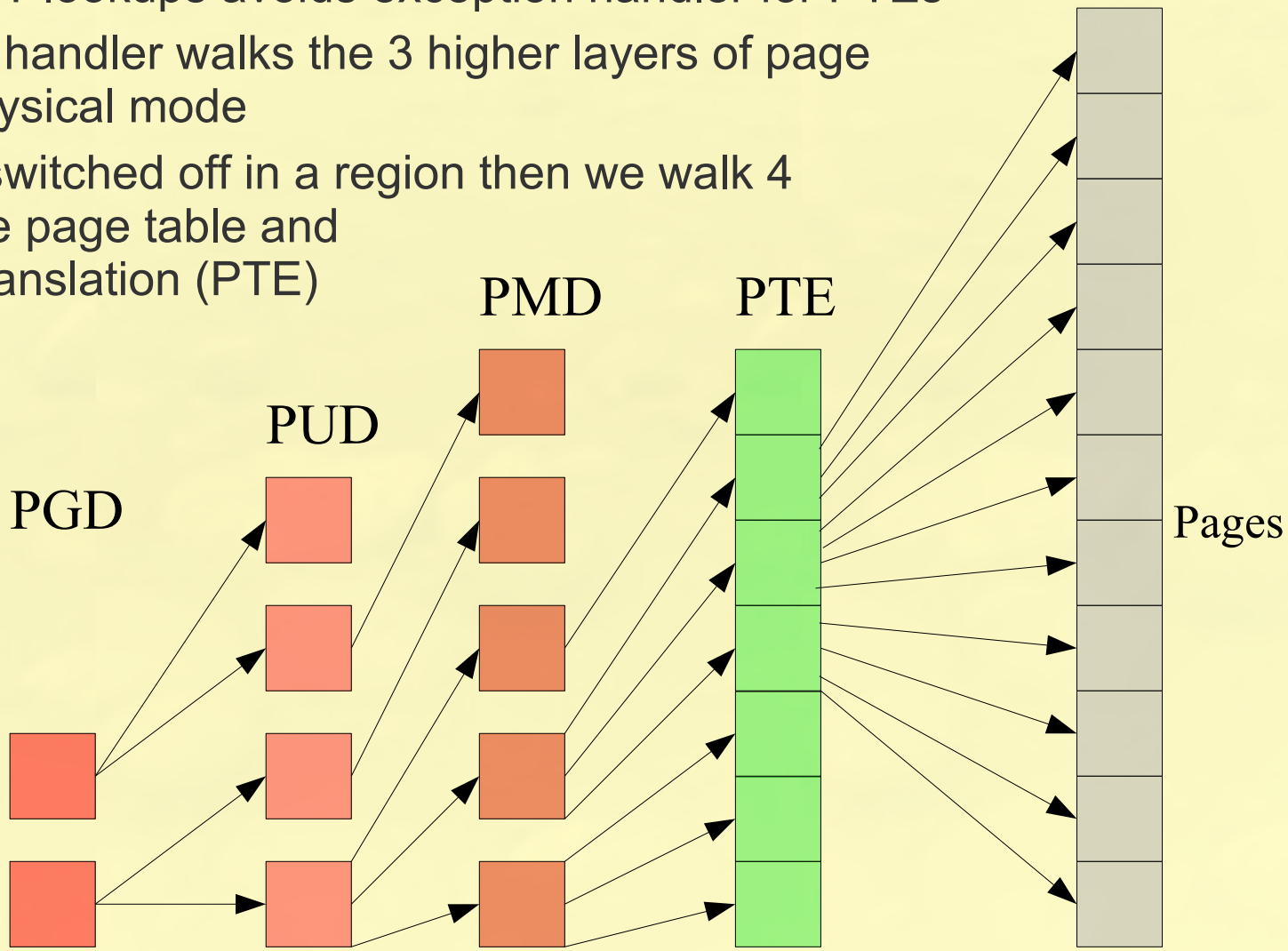
Memory Regions under Linux

- Regions
  - Segment of memory space
  - Different mappings
  - Page sizes
  - VHPT walker
  - Per cpu area has different page size

Rgn	PageSize	VHPT	Purpose
0	?	?	Was used for IA32 emulation (broken?)
1	16k	On	User space: Executable pages
2	16k	On	User space: Data
3	16k	On	User space: Stack
4	256m	On	Huge pages with configurable page size
5	16k	On	Gate area, Kernel VMALLOC and virtual memmap
6	16/64m	Off	Uncached 1-1 mapping for I/O
7	16/64m	Off	1-1 mapping for kernel data area. Per cpu area

Linux Page Table Emulation

- 16k page sized chunks containing 512 pointers each
- Use of VHPT lookups avoids exception handler for PTEs
- VHPT miss handler walks the 3 higher layers of page tables in physical mode
- If VHPT is switched off in a region then we walk 4 layers of the page table and insert the translation (PTE) ourselves.



Linux/IA64 Page Tables continued

- 3 or 4 level page table support
- 3 levels support up to 16TB of memory
  (2^(3*11) *PAGE_SIZE = 2^(33+14) = 2^48
  3 bits are gone for region ID and we need some space for the VMLPT. -> ~ 2^44 = 16TB
- 4 levels support up to ~1PB
  2^(44+14) = 58 bits, 3 bits for region which would leave 54 but we only seem to need 50?
  (limit imposed by SPARSEMEM)

Virtual Memory Map

- Placed in region 5. Thus 16k page size.
- A 4GB node needs ~16M virtual memmap
- 1024 TLB entries via VHPT lookups
- 2 via VHPT miss handler

TLB Performance Issues

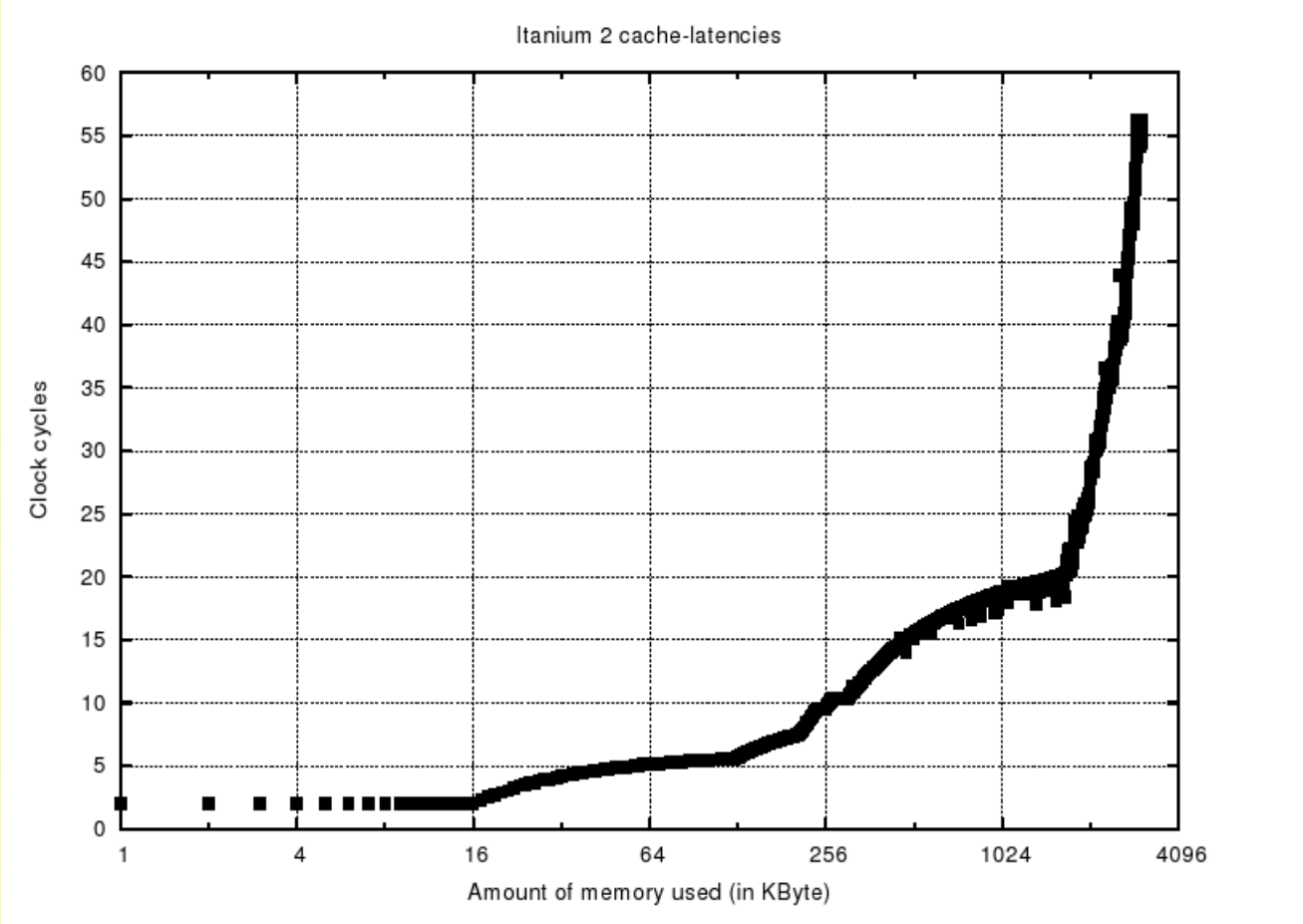
- L1/L2 TLB handler L1. Each maps 4k. We have 32 which can therefore only map 128k of memory.
- 128 L2/TLB entries. Page size 16K -> 2M memory w/o L2 TLB fault.
- Paper “Accessing Data on SGI Altix: An Experience with Reality” by Juckeland, Mueller, Nagel and Pflueger, TU Dresden (http://www.cs.utah.edu/wmpi/2006/final-version/wmpi-posters-3-Juckeland.pdf).
- Paper investigates the issue for user space. But this is applicable to general use of 16k pages in the kernel.
- Virtual memmap using 16k pages. A 4GB node needs 16M for the virtual memory map. Sparse access -> potential for TLB pressure.

ITANIUM 2 TLB CHARACTERISTICS [1]

Characteristic	Instruction TLB		Data TLB	
	Level 1	Level2	Level 1	Level2
Number of entries	32	128	32	128
Associativity	Full		Full	
Penalty for L1 miss	2 clock cycles		4 clock cycles	

Itanium Madison TLB issues with 16K page at 2M

- Fast access to 16k L1 memory
- L2 kicks in and L2 TLBs are used
- At 2M VHPT lookups etc hit us and performance deteriorates rapidly.
- Itanium in current configuration unable to utilize all of the L3 cache effectively due to TLB pressure!
- It is important for the kernel to avoid TLB use
- 16K pages for memory map used
- A 4G nodes memory map is 16M
- If the kernel does intensive sparse access to page structs the we are likely run into a lot of TLB pressure.



Larger Page Sizes on IA64

- Necessary to avoid TLB pressure
- But we cannot have variable page sizes due to short VHPT format.
- Region 6 and 7 do not use VHPT walker but every TLB is generated by exception handler.
- Thus one can do custom page sizes in Region 6 and 7
- Per cpu area already has custom size
- We have a couple of unused address. Bit 60 and below
- Exception handlers could be programmed to use these bits to generate custom page size TLBs

A programmable TLB generator

- Use unused higher address bits
- Enable / Disable page table using bit 60
- 53 lower bits stay
- Bits 54-59 become page size bits
- All IA64 MMU page sizes supported
- 4k, 8k, 16k, 64k, 256k, 1M, 4M, 16M, 64M, 256M, 1G

- region7_pgdir is a new pgd for variable kernel pages.
- Segmented into 8 sections to support multiple page sizes
- Only usable for the Kernel access
- Large structures can be mapped with larger TLB
- VHPT off -> all TLB misses do physical walks.

Conclusion

- Best to avoid small page sizes like 16k.
- 64k page size for kernel?
- Use larger page size for virtual memmap to avoid the issue for the kernel.
- VKP Patchset posted October 2006 on Linux-IA64 mailing list.
- Integrating VMMEMMAP into SPARSEMEM
  - Uses simplified version with only a single bit
  - Two page sizes only 16K/16M.
  - We want 4M version? Would require 2 stage lookup
- Questions?
- For further information contact clameter@sgi.com.